

УДК: 61:004.45

**АСПЕКТИ РЕАЛІЗАЦІЇ ІНТЕГРОВАНОГО СЕРЕДОВИЩА ПЕРЕВІРКИ ЗНАНЬ
У МЕДИЧНІЙ ОСВІТІ****А.В. Семенець***Тернопільський державний медичний університет імені І.Я.Горбачевського*

Метою даної роботи є практична реалізація бази завдань, а також розробка алгоритму автоматичного проектування тест-білетів. Запропоновано інформаційну модель бази даних для зберігання тестових завдань, що практично реалізована в СУРБД Firebird. Розглянуто загальні питання проектування тестів. Розроблено алгоритм автоматичного проектування тест-білетів, що реалізований мовою Object Pascal з використанням інструментального середовища Delphi.

Ключові слова: медична освіта, інформаційна модель, підготовка тестування, проектування тест-білета, база даних, розробка програмного забезпечення.

**АСПЕКТЫ РЕАЛИЗАЦИИ ИНТЕГРИРОВАННОЙ СРЕДЫ ПРОВЕРКИ ЗНАНИЙ
В МЕДИЦИНСКОМ ОБРАЗОВАНИИ****А.В. Семенец***Тернопольский государственный медицинский университет имени И.Я.Горбачевского*

Целью данной работы является практическая реализация базы заданий, а также разработка алгоритма автоматического проектирования тест-билетов. Предложена информационная модель базы данных для хранения тестовых заданий, которая практически реализована в СУРБД Firebird. Рассмотрены общие вопросы проектирования тестов. Разработан алгоритм автоматического проектирования тест-билетов, который реализован на языке Object Pascal с использованием инструментальной среды Delphi.

Ключевые слова: медицинское образование, информационная модель, подготовка тестирования, проектирование тест-билета, база данных, разработка программного обеспечения.

**ON IMPLEMENTATION OF THE TEST KNOWLEDGE INTEGRATED
ENVIRONMENT IN THE MEDICAL EDUCATION****A.V. Semenets***Ternopol medical state university named I.Ya.Gorbachevsky*

The purpose of this work is the practical implementation of the test database and also development of the algorithm for the automatic test sets construction. Information model of the database for the test storage is offered. This model was practically implemented using the Firebird database managment system. Basic topics of the test sets development are considered. The algorithm for the automatic test sets construction was developed. This algorithm was implemented in terms of Object Pascal programming language using the Delphi development system.

Keywords: medical education, information model, test preparing, test sets development, databases, application development.

Вступ. Проблема якісного контролю знань особливо актуальна для медичної освіти. Одним з інструментів для одержання керуючої педагогічної інформації є результати тестування.

У порівнянні із традиційними формами контролю (напр. іспити) тести часто виявляються більш об'єктивним й якісним способом контролю. Результат стандартизованого тестування дозволяє, до того ж, зіставити рівень окремого об'єкта (студента, групи)

© А.В.Семенець

по предмету в цілому (або по окремих темах) із середнім рівнем або з подібним об'єктом.

Однак, часто замість повноцінних тестів використовуються тестові завдання – набір питань із варіантами відповідей, які, можливо, і корисні, але не є тестами по своїй суті. Це призводить суть тестування до простого опитування й дискредитує сам метод тестування, хоча в закордонній практиці він визнаний одним з найбільш

надійних засобів масового контролю досягнень абітурієнтів [1].

На початку 20-го століття були закладені основи теорії тестування, які активно розвивалися до початку 70-х років. В 1968 році Ф. Лорд і М. Новик [2, 3] сформулювали основні постулати математичної моделі класичної теорії тестування. При цьому, під тестом *T* розуміється структурована система завдань і, відповідна їй, процедура перевірки цих завдань, що забезпечує однозначність інтерпретації отриманих результатів тестування.

У зв'язку зі зростаючим використанням сучасної комп'ютерної техніки при визначенні рівня підготовки абітурієнтів, і її широким впровадженням у практику роботи освітніх організацій виникає завдання формалізації процедур і методів підготовки тестових завдань, створення технології тестування, розрахованої на масового користувача.

Ціль даної роботи – представити одну практичну реалізацію бази завдань (бази даних), а також алгоритму автоматичного проектування тест-білетів.

У Тернопільському державному медичному університеті імені І. Я. Горбачевського (ТДМУ) у цей момент проводиться експеримент по впровадженню комплексного тестування як одного з методів контролю знань студентів. Для технічного забезпечення підготовки й проведення тестування на кафедрі ме-

дичної інформатики розробляється інтегроване середовище перевірки знань у медичній освіті (ІСПЗМО). Дана система відноситься до інформаційно-керуючих систем (ІКС) [4].

Як інструментальні засоби для технічної реалізації ІСПЗМО були обрані [5]:

- мова розробки додатку – Java;
- середовище розробки додатку – Java Studio Creator – вільно розповсюджуване програмне забезпечення;
- програмний сервер управління реляційними базами даних (СУРБД) – Firebird – вільно розповсюджуване програмне забезпечення;
- середовище виконання додатку – Java Application Server – вільно розповсюджуване програмне забезпечення;

1. Структура бази завдань

Основою для створення тест-білетів служить база завдань. При розробці структури бази завдань виникає питання про те, які параметри завдань повинні бути описані. Якщо при проектуванні тест-білету використовуються процедури класичної теорії тестування, то в якості основних (статистичних) параметрів повинні бути вказані *коефіцієнт вирішуваності k_i* і *коефіцієнт селективності D_i* . У даній роботі ми спиралися на структуру, запропоновану в [1]:

Найменування завдання	Коефіцієнт вирішуваності k_i	Коефіцієнт селективності D_i	Коефіцієнт вирішуваності дистракторів			
			$k_i(1)$	$k_i(2)$	$k_i(\dots)$	$k_i(x)$

Початкові значення коефіцієнтів вирішуваності k_i і селективності D_i визначаються за результатами пробного тестування, і уточнюються в процесі експлуатації тестового завдання.

Нехай за результатами тестування отримана матриця відповідей *D* розміром *NxL* (*N* випробуваних, *L* завдань), причому зроблено вибракування рядків і стовпців, що цілком складаються з нулів й одиниць. Знаходження статистичних класичних параметрів проводиться по формулах [1]:

1. Коефіцієнт вирішуваності завдання

$$k_i = \frac{1}{N} \sum_{n=1}^N u_{ni} \left(i = \overline{1, n} \right)$$

(1.1)

де u_{ni} - оцінка виконання *i*-го завдання *n*-м випробуваним.

1. Коефіцієнт селективності завдання

$$D_i = k'_i - k''_i$$

(1.2)

де k'_i – коефіцієнт вирішуваності *i*-го завдання кращої половини тестованих;

k''_i – коефіцієнт вирішуваності *i*-го завдання гіршої половини тестованих.

Класичні параметри істотно залежать від популяції тестованих, на яких була отримана статистика. При розробці тест-білетів класичні параметри зручні на етапі первинного конструювання тест-білета, або коли база завдань використається приблизно для тієї ж категорії випробуваних, на яких проводилося калібрування завдань (при стабільному навчальному процесі – у школі, вузі й т. д). Ця причина й обумовила вибір саме класичних параметрів тестування для використання їх у рамках розроблюваної ІСПЗМО.

В експериментальному проекті ІСПЗМО вищевказана структура бази завдань реалізована у вигляді двох таблиць реляційної бази даних [1], які мають вигляд, представлений на рисунку 1:

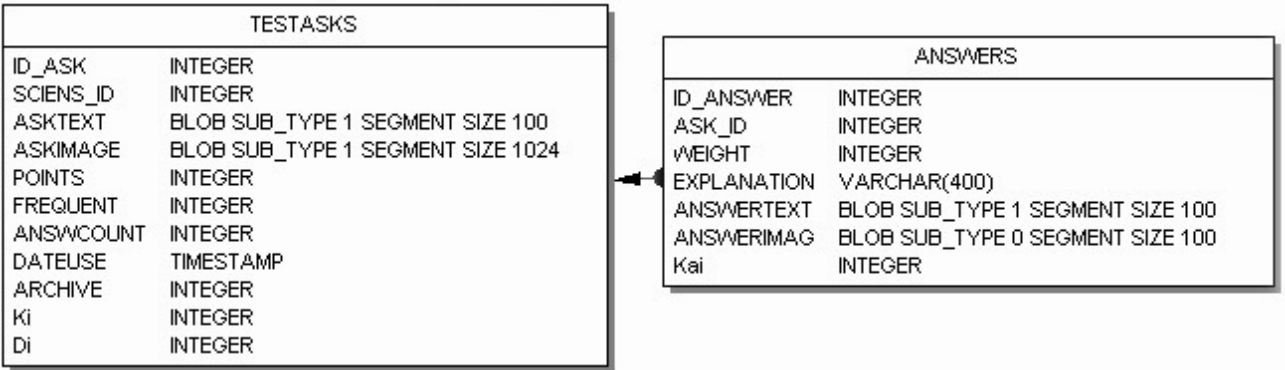


Рис. 1. Структура бази завдань в експериментальному проєкті ІСПЗМО.

Дана структура має кілька переваг:

1. Застосування полів типу **BLOB** для зберігання текстів питань і відповідей дозволяє використовувати розширене форматування різноманітні шрифти, ефекти, кольорове оформлення.
2. Поле **POINTS** служить для попереднього задання рівня складності питання викладачем, і, отже, дозволяє створювати тести з питаннями різної складності.
3. Поле **ARCHIVE** забезпечує можливість тимчасового виключення питання з переліку використовуваних для проектування тест-білета, причому, без видалення з бази даних.
4. Завдяки відношенню типу «один-до-багатьох» між таблицею питань **TESTASKS** і таблицею відповідей **ANSWERS** є можливість для одного питання ввести будь-яку кількість відповідей [6].
5. Наявність поля **WEIGHT** дозволяє створювати тестові питання з декількома правильними відповідями різної ваги.

2. Загальна схема проектування тестів й аспекти її реалізації в ІСПЗМО

Тут ми опиралися на основні етапи створення тестових матеріалів, запропоновані в [1]. Вони полягають у наступному:

1. Постановка цілей педагогічних вимірів індивідуальна або масова діагностика, завдання відбору, селекції або оцінки ефективності освітньої програми (методики навчання). Метою педагогічного експерименту в ТДМУ є масова діагностика семестровий контроль знань студентів.
2. Визначення й вибір підходящої моделі проектування тестів. Метою педагогічного експерименту в ТДМУ є оцінка рівня підготовленості тобто проєктовані тести повинні містити завдання з найбільшою кількістю навчальних тем. Статистичний розподіл результатів вимірів у цьому випадку близький до нормального.
3. Підготовка банку відкаліброваних завдань. Найбільш трудомісткий етап у процедурі розробки тестів. У розробленому експериментальному проєкті **ІСПЗМО** для цих цілей використовуються СУРБД Firebird і веб-редактор, що працює під управлінням Java Application Server* . Для первинного наповнення банку тестових завдань знадобився навчальний рік (2006/2007 н.р.)
4. Створення структури тестів (переліку контрольованих навчальних елементів). У проєкті **ІСПЗМО** створення структури здійснюється вручну за допомогою програмного комплексу з веб-інтерфейсом [1].
5. Попереднє визначення валідності. Після створення структури можна експертно визначити змістовну валідність тестів. Під валідністю розуміється ступінь досягнення тестами поставлених цілей і відсутністю факторів, що спотворюють результати виміру. Звичайно проводиться вручну, групою експертів-викладачів.
6. Проектування тестів, генерація необхідної кількості варіантів. Найбільш важливий етап тестування. У даній версії **ІСПЗМО** для відпрацювання алгоритмів використовується утиліта, написана на мові Object Pascal з використанням інструментального середовища Delphi. Головні особливості алгоритму генерації варіанту тесту будуть розглянуті нижче.
7. Рецензування й редагування тестових матеріалів є наступним етапом перевірки якості тестових завдань. Воно може проходити у формі простого обговорення членами робочої групи й у формі зовнішнього (незалежного) рецензування.
8. Пробне тестування. Воно покликано визначити валідність і надійність тесту, відбракувати невдалі завдання. Пробне тестування проводиться, як правило, на відносно невеликих групах для того, щоб:
А) визначити первинні статистичні характеристики завдань і матеріалів у цілому,
В) виконати коректування завдань.
9. Аналіз і корекція. Після пробного тестування може виявитися, що отримані результати не (по-

* Більш детальна інформація про експериментальний проєкт ІСПЗМО представлена на <http://www.tdmu.edu.te.ua/departments/informatics>

вністю) відповідають поставленим цілям. У цьому випадку необхідне коректування структури тестових матеріалів і набору завдань.

3. Реалізація базового алгоритму проектування тест-білетів

Розглянемо деякі загальні технічні питання, які виникають при необхідності автоматично спроектувати тест-білет:

1. Необхідна повна реалізація структури тесту – повинні бути підготовлені питання по кожному розділу тест-білету.

1. Кількість розділів і питань по кожному розділу повинно відповідати заданим.

2. Вибірка питань повинна відбуватися випадковим чином. На етапі пробного тестування частота використання питань у тест-білетах повинна бути рівномірною.

3. Структура квитка повинна бути збережена, для наступного аналізу й корекції.

Спрощена блок-схема розробленого авторами алгоритму автоматичного проектування тест-білетів показана на рис 2. На даному етапі розробки ІСПЗМО

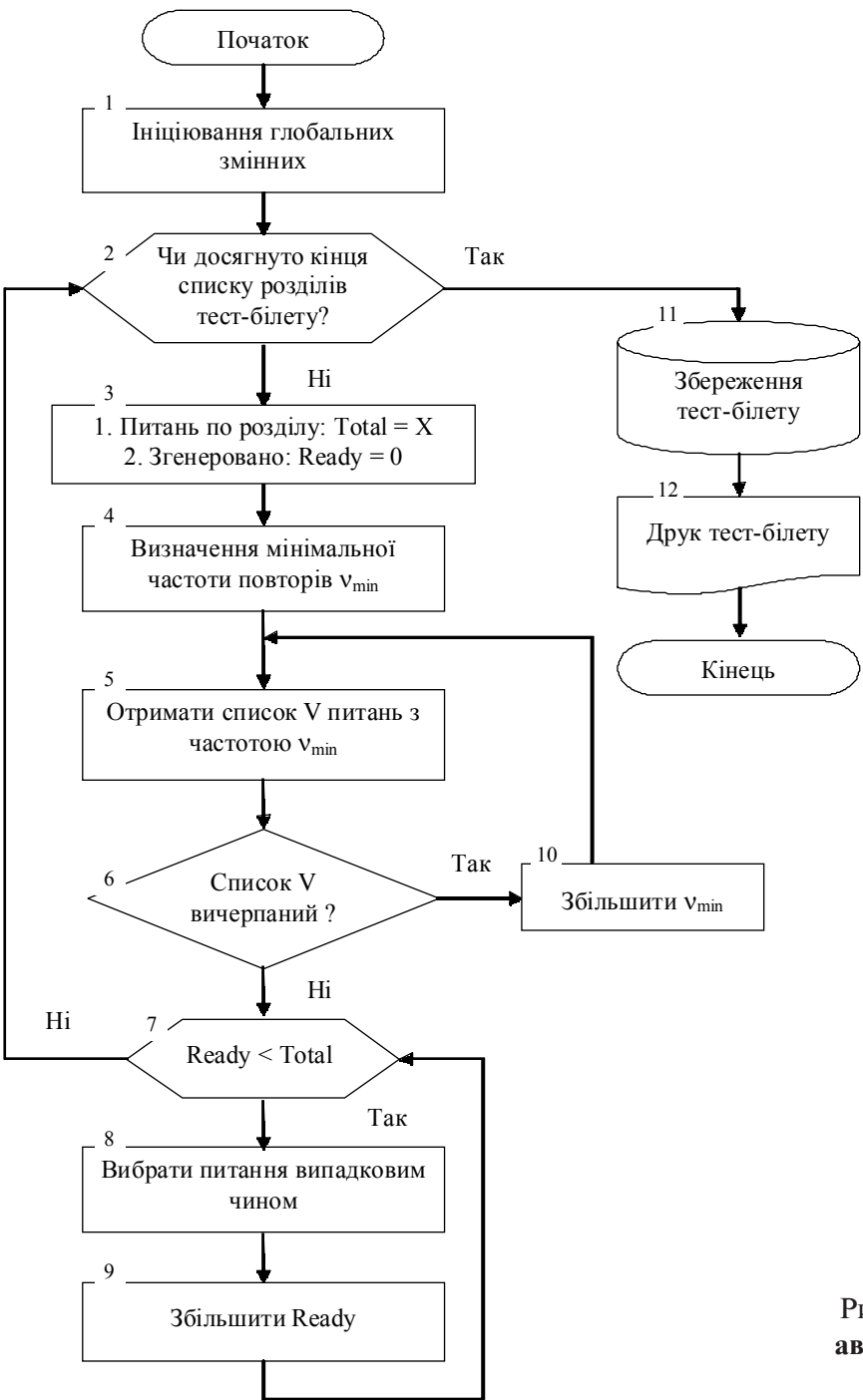


Рис. 2. Блок-схема алгоритму автоматичного проектування тест-білетів

даний алгоритм реалізований мовою Object Pascal з використанням інструментального середовища Delphi. Після остаточного відпрацювання алгоритм планується переписати на Java для використання у вигляді сервлета під керуванням Java Application Server [1].

Зазначений на рис 2 алгоритм працює в такий спосіб: на кроці 1 ініціалізуються глобальні змінні даного модуля. На кроці 2 у циклі обробляються всі розділи тест-білета. На кроці 3 установлюються початкові умови циклу: кількість питань по розділу Total і кількість вже згенерованих питань Ready. Крок 4 використовується для визначення мінімальної частоти повторів питань у базі nmin. Далі (крок 5) з бази даних вибирається список V питань із частотою nmin. Наступною йде перевірка (крок 6) – чи вичерпаний список V. У випадку позитивної відповіді – частота збільшується (крок 10) і проводиться добування нового списку. Якщо ж список не вичерпаний, перевіряється умова – згенеровано чи ні належне число питань (крок 7). Після цього випадковим чином вибирається питання зі списку V (крок 8). Кількість готових питань Ready збільшується (крок 9) і відбувається повернення до початку циклу генерації (крок 7). Якщо ж згенеровано задане число питань (Ready = Total), відбувається повернення в головний цикл – до кроку 2. У випадку, коли всі розділи квитка оброблені, тест-білет зберігається в базі даних (крок 11), і друкується (крок 12).

4. Проблема проектування тест-білетів

Комп'ютерна генерація тесту вимагає розробки математичного апарату, що формально представляє різні проблеми конструювання тестів у вигляді завдань математичного програмування. У роботах, присвячених даній проблемі [1], розглядаються підходи, що дозволяють формалізувати завдання конструювання тест-білета як завдання цілочисельного математичного про-

грамування: знайти рішення, що забезпечує оптимальне (максимальне або мінімальне) значення цільової функції при заданих обмеженнях на множині можливих значень рішення, при цьому координати вектора рішення передбачаються двозначними – 0 або 1.

Зіставляючи з тест-білетом вектор x із цілочисельними координатами $x_i = 0$ v 1 , значення яких визначають, включено чи ні завдання з номером i значення, її чисельно оцінювали необхідну якість тест-білету (наприклад, кількість використовуваних завдань), а також задаючи обмеження $f_j(x) \geq 0$, наприклад, на множину використовуваних завдань, приходимо до наступної задачі цілочисельного програмування:

$$\min \{ \Phi_0(x) / f_j(x) \geq 0, j=1, \dots, S; x_i = 0 \text{ or } 1, i=1, \dots, N \} \quad (1.3)$$

Разом з тим, тільки найпростіші випадки описуються за допомогою однієї цільової функції, значення якої вимірюють якість вирішення тест-білету. У більш складних ситуаціях якість вирішення не може бути оцінена єдиною цільовою функцією. Такий стан справ при роботі в багатокритеріальних ситуаціях, наприклад, коли конструктор повинен враховувати якість декількох параметрів тест-білета [1, 9].

Висновки. У роботі розглядаються питання практичної реалізації бази завдань, а також розробки алгоритму автоматичного проектування тест-білетів. Запропоновано інформаційну модель бази даних для зберігання тестових завдань, що практично реалізована в СУРБД Firebird. Розглянуто загальні питання проектування тестів. Розроблено алгоритм автоматичного проектування тест-білетів, що реалізований мовою Object Pascal з використанням інструментального середовища Delphi. Метою наступних досліджень повинна бути методика автоматичного аналізу й корекції спроектованих тест-білетів з використанням зазначеного математичного апарату й результатів пробного тестування.

Література

1. Бульгин В.Г. Основы автоматизации процесса обучения / В.Г. Бульгин. - Йошкар-Ола, 2003. - 190 с.
2. Lord P.M., Novic M.R. Statistical Theories of Mental Test Scores. Reading, Mass.: Addison-Wesley, 1968.
3. Novic M.R. The Axioms and Principle Results of Classical Test Theory / Journal of Mathematical Psychology, 1966, N3. - P. 1-18.
4. Энциклопедия кибернетики: В 2 т. – К.: Гол. ред УРЕ, 1974.
5. Х.М. Дейтел, П. Дж. Дейтел, С.И. Сантри Технологии программирования на Java 2: В 3 т. Пер. с англ. – М.: ООО “Бином-пресс”, 2003 г.
6. Марценюк В.П. Медицинская информатика. Проектирование и использование баз данных. – Тернополь: Укр-

медкнига, 2001. – 178 с.

7. Марценюк В.П., Кравец Н.О. О программной среде проектирования интеллектуальных медицинских баз данных // Клиническая информатика и телемедицина - 2004, №1. – С. 47-53.
8. Ткачук Н.В. Концепция интегрированной среды реинжиниринга сложных информационных управляющих систем // Проблемы информатики и управления. – 2003. - №1. – С. 74-83.
9. Наводнов В.Г., Петропавловский М.В., Ельцын А.В. Автоматизированное проектирование педагогических измерительных материалов: Препринт N 2/97. - Йошкар-Ола. Научно-информационный центр государственной аккредитации, 1997.